

מדעי המחשב

על פי תכנית הרפורמה ללמידה משמעותית

הוראות לנבחן

א. משך הבחינה: שלוש שעות.

ב. מבנה השאלון ומפתח ההערכה: בשאלון זה שלושה פרקים.

פרק ראשון – בפרק זה שלוש שאלות, ענה על פי

ההוראות בפרק. — $(10 \times 1) + (15 \times 1)$ — 25 נקודות

פרק שני – בפרק זה שלוש שאלות,

ומהן עליך לענות על שתיים. — (25×2) — 50 נקודות

פרק שלישי – בפרק זה שאלות בארבעה מסלולים שונים.

ענה על שאלה אחת במסלול שלמדת. — (25×1) — 25 נקודות

סה"כ — 100 נקודות

ג. חומר עזר מותר בשימוש: כל חומר עזר, חוץ ממחשב הניתן לתכנות.

ד. הוראות מיוחדות:

(1) את כל התכניות שאתה נדרש לכתוב בשפת מחשב בפרקים הראשון והשני

כתוב בשפה אחת בלבד — Java או C#.

(2) רשום על הכריכה החיצונית של המחברת באיזו שפה אתה כותב — Java או C#.

(3) רשום על הכריכה החיצונית של המחברת את שם המסלול שלמדת.

המסלול הוא אחד מארבעת המסלולים האלה:

מערכות מחשב ואסמבלי, מבוא לחקר ביצועים, מודלים חישוביים, תכנות מונחה עצמים.

הערה: בתכניות שאתה כותב לא יורדו לך נקודות, אם תכתוב אות גדולה במקום

אות קטנה או להפך.

כתוב במחברת הבחינה בלבד, בעמודים נפרדים, כל מה שברצונך לכתוב בטייטה (ראשי פרקים, חישובים וכדומה).

רשום "טייטה" בראש כל עמוד טייטה. רישום טייטות כלשהן על דפים שמחוץ למחברת הבחינה עלול לגרום לפסילת הבחינה!

ההנחיות בשאלון זה מנוסחות בלשון זכר ומכוונות לנבחנות ולנבחנים כאחד.

בהצלחה!

/המשך מעבר לדף/

השאלות

בשאלון זה שלושה פרקים.

עליך לענות על שאלות משלושת הפרקים, לפי ההוראות בכל פרק.

פרק ראשון (25 נקודות)

הערה: בכל שאלה שנדרשת בה קליטה, אין צורך לבדוק את תקינות הקלט.

לפותרים ב-Java: בכל שאלה שנדרשת בה קליטה, הנח שבתכנית כתובה ההוראה:

```
Scanner input = new Scanner (System.in);
```

ענה על שאלה 1 – חובה (10 נקודות)

1. כתוב פעולה חיצונית big ב-Java או Big ב-C# שתקבל מערך של מספרים שלמים. הפעולה תחזיר את המציין (האינדקס) של האיבר שהערך שלו הוא הגדול ביותר במערך. אם המספר מופיע יותר מפעם אחת במערך, הפעולה תחזיר את המציין (האינדקס) הקטן ביותר מבין המופעים.

ענה על אחת מן השאלות 2-3 (לשאלה – 15 נקודות).

2. בהיערכות לקראת משחקי ספורט בין-לאומיים הגדירו מחלקה **Game** שיש לה שלוש תכונות: שם המשחק – `gameName` מטיפוס מחרוזת, מספר השחקנים – `numPlayers` מטיפוס שלם, אם המשחק מתקיים במים – `isWater` מטיפוס בוליאני. הנח שלכל תכונה הוגדרו ב-Java פעולות `get` ו-`set` וב-C# פעולות `Get` ו-`Set`.
 - א. כתוב ב-Java או ב-C# במחלקה **Game** פעולה בונה שתקבל ערכים בעבור כל תכונה.
 - ב. הגדירו מחלקה מדינה – **Country** שיש לה שתי תכונות: שם המדינה – `countryName` מטיפוס מחרוזת, מערך חד-ממדי `games` בגודל 43 מטיפוס **Game**. המערך מכיל את המשחקים שהמדינה משתתפת בהם. הנח שלכל תכונה הוגדרו ב-Java פעולות `get` ו-`set` וב-C# פעולות `Get` ו-`Set`.
 - (1) כתוב ב-Java או ב-C# את כותרת המחלקה **Country** ואת התכונות שלה.
 - (2) כתוב ב-Java או ב-C# במחלקה **Country** פעולה בונה שתקבל את שם המדינה. הפעולה תבנה עצם בעבור מדינה שאינה משתתפת באף לא משחק אחד.
 - (3) כתוב ב-Java או ב-C# במחלקה **Country** פעולה בוליאנית שתקבל את שם המשחק ותחזיר `true` – אם המדינה משתתפת במשחק, אחרת – הפעולה תחזיר `false`.

3. לפניך קטע תכנית הכתוב ב־ Java וב־ C#

```

Java
int i , j;
for (i = 1; i <= n; i++)
{
    a[i] = i;
}
for (i = 2; i < n; i++)
{
    if (a[i] != 0)
    {
        for (j = i + 1; j <= n; j++)
        {
            if (a[j] % a[i] == 0)
            {
                a[j] = 0;
            }
        }
    }
}

for (i = 1; i <= n; i++)
{
    if (a[i] != 0)
    {
        System.out.println(a[i]);
    }
}

```

```

C#
int i , j;
for (i = 1; i <= n; i++)
{
    a[i] = i;
}
for (i = 2; i < n; i++)
{
    if (a[i] != 0)
    {
        for (j = i + 1; j <= n; j++)
        {
            if (a[j] % a[i] == 0)
            {
                a[j] = 0;
            }
        }
    }
}

for (i = 1; i <= n; i++)
{
    if (a[i] != 0)
    {
        Console.WriteLine(a[i]);
    }
}

```

א. רשום מה יודפס עבור $n = 15$.

ב. רשום במשפט אחד מה מבצע קטע התכנית, כלומר נסח את הבעיה

שהפתרון שלה הוא קטע התכנית.

פרק שני (50 נקודות)

שים לב: בכל שאלה שנדרש בה מימוש אתה יכול להשתמש בפעולות של המחלקות תור, מחסנית, עץ בינרי וחוליה, בלי לממש אותן. אם אתה משתמש בפעולות נוספות, עליך לממש אותן.

ענה על שתיים מן השאלות 4-6 (לכל שאלה – 25 נקודות).

4. לפניך הגדרה של חמש פעולות הפועלות על מבנה נתונים כלשהו.
- שים לב:** שמות הפעולות שלפניך אינם כתובים ב־ Java או ב־ C#.
- `insert(x)` פעולה המכניסה למבנה איבר שערכו x מטיפוס שלם.
 - `showMin()` פעולה המחזירה את הערך הנמוך ביותר במבנה, בלי לשנות את המבנה.
 - `getMax()` פעולה המחזירה את האיבר שערכו הוא הגדול ביותר במבנה, ומוציאה אותו מן המבנה. אם יש יותר מאיבר אחד כזה, הפעולה תחזיר ותוציא את זה שמופיע ראשון.
 - `exists(x)` פעולה בוליאנית המחזירה `true` אם האיבר שערכו x קיים במבנה. אחרת – הפעולה מחזירה `false`.
 - `div7()` פעולה בוליאנית המחזירה `true` אם קיים במבנה איבר שערכו מתחלק ב־ 7 בלי שארית. אחרת – הפעולה מחזירה `false`.
- נרצה להציע מבני נתונים העומדים בדרישות סיבוכיות שונות למימוש פעולות מתוך חמש הפעולות שהוגדרו.
- דוגמה: רוצים להציע מבנה נתונים שאפשר לבצע עליו את הפעולות `insert` , `showMin` , בסיבוכיות $O(1)$, ואת הפעולות `exists` , `getMax` בסיבוכיות $O(n)$. לשם כך נגדיר את מבנה הנתונים ונסביר כיצד ימומשו הפעולות.
- שים לב: במבנה זה אין צורך להתייחס לפעולה `div7` .

מבנה נתונים מתאים מורכב מ:

- רשימה מקושרת דו-כיוונית, lst מטיפוס שלם.
- מצביע על האיבר המינימלי שהוכנס לרשימה, min.

הפעולות יבוצעו כך:

הפעולה	הסבר כיצד תמומש	נימוק מדוע המימוש עומד בדרישות הסיבוכיות
insert(x)	— הכנסת האיבר x לראש הרשימה. — אם האיבר קטן מן המינימום עד כה, עדכון המצביע על min כך שיצביע על האיבר החדש.	— הכנסת איבר לראש הרשימה — $O(1)$. — אם הוכנס איבר שקטן מן המינימום: עדכון המצביע לאיבר המינימלי — $O(1)$. — אם הוכנס איבר ראשון — הכנסה לראש הרשימה ב- $O(1)$ ועדכון המצביע לאיבר הראשון שהוא גם המינימלי. סך הכול — $O(1)$.
showMin()	החזרת הערך של האיבר שעליו מצביע min.	החזרת הערך — $O(1)$.
exists(x)	מעבר על הרשימה lst וחיפוש האיבר שערכו x.	במקרה הגרוע — מעבר על כל הרשימה — $O(n)$.
getMax()	מעבר על הרשימה lst, חיפוש האיבר שערכו מקסימלי והוצאתו מן הרשימה.	— חיפוש האיבר שערכו מקסימלי $O(n)$. — הוצאתו מן הרשימה $O(1)$.

לפניך שני סעיפים א-ב. בעבור כל אחד מן הסעיפים עליך להציע מבנה נתונים מתאים העומד בדרישות המפורטות בסעיף. המבנה יכול להיות מורכב משילוב של כמה מבנים וטיפוסים שלמדת. לכל אחת מן הפעולות הסבר כיצד תממש אותה, ונמק מדוע המימוש עומד בדרישות (כפי שהוצג בטבלה שבדוגמה). אין צורך לממש את הפעולות.

א. ביצוע הפעולות insert, exists בסיבוכיות $O(n)$, וביצוע הפעולות showMin, getMax בסיבוכיות $O(1)$.

ב. ביצוע הפעולות insert, getMax בסיבוכיות $O(n)$, וביצוע הפעולה div7 בסיבוכיות $O(1)$.

5. לפניך תכנית הכתובה ב־Java וב־C#. הפעולה what ב־Java ו־What ב־C# מקבלת מערך חד־ממדי המכיל מספרים שלמים, ומספר שלם k , $k > 0$. כל המספרים במערך הם בין 0 ל־ k (כולל).

Java

```
public class Program
{
    public static int[] what(int[] arr, int k)
    {
        int n = arr.length;
        int[] b = new int[n];
        int[] c = new int[k+1];

        for (int i = 0; i < k+1; i++) c[i] = 0;
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;    /*

        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i-1];    /**

        for (int j = n-1 ; j >= 0; j--)                          ****
        {
            b[c[arr[j]] - 1] = arr[j];
            c[arr[j]] = c[arr[j]] - 1;
        }
        return b;
    }

    public static void main(String[] args)
    {
        int[] arr = new int[] {5,0,2,1,3,0,5};
        arr = what(arr, 5);
        for (int i = 0; i < arr.length; i++) System.out.println(arr[i]);
    }
}
```

C#

```
public class Program
{
    public static int[] What(int[] arr, int k)
    {
        int n = arr.Length;
        int[] b = new int[n];
        int[] c = new int[k+1];

        for (int i = 0; i < k+1; i++) c[i] = 0;
        for (int j = 0; j < n; j++) c[arr[j]] = c[arr[j]] + 1;    /*

        for (int i = 1; i < k+1 ; i++) c[i] = c[i] + c[i- 1];    /**

        for (int j = n- 1 ; j >= 0; j--)                          /****
        {
            b[c[arr[j]] - 1] = arr[j];
            c[arr[j]] = c[arr[j]] - 1;
        }
        return b;
    }

    public static void Main(string[] args)
    {
        int[] arr = new int[] {5,0,2,1,3,0,5};
        arr = What(arr, 5);
        for (int i = 0; i < arr.Length; i++) Console.WriteLine(arr[i]);
    }
}
```

א. (1) סרטט את המערך c לאחר ביצוע הלולאה המסומנת /*

(2) סרטט את המערך c לאחר ביצוע הלולאה המסומנת /**

(3) עקוב אחר ביצוע הלולאה המסומנת /****

במעקב עליך להראות את: j , arr[j] , c[arr[j]] , b[c[arr[j]] - 1] , המערך b והמערך c.

ב. מה מבצעת הפעולה what ב-Java או What ב-C# ?

ג. מהי סיבוכיות זמן הריצה של הפעולה what ב-Java או What ב-C# ? נמק את תשובתך.

/המשך בעמוד 8/

6.

א.

ממש פעולה חיצונית ב־ Java או Exist ב־ C#. הפעולה תקבל עץ בינרי t מטיפוס שלם ומספר שלם x. הפעולה תחזיר true אם יש בעץ צומת שערכו x, אחרת – הפעולה תחזיר false. אם העץ ריק – הפעולה תחזיר false.

ב.

לפניך הפעולה ב־ Java check(t1, t2) וב־ C# Check(t1, t2). הפעולה מקבלת שני עצים בינריים לא ריקים מטיפוס שלם, t1 ו־ t2, ומחזירה רשימה המכילה את כל המספרים הנמצאים בעץ t1 ואינם נמצאים בעץ t2. הפעולה מזמנת פעולה נוספת המקבלת שלושה פרמטרים.

Java

```
public static Node<Integer> check(BinNode<Integer> t1, BinNode<Integer> t2)
{
    Node<Integer> first = new Node<Integer> (- 1);
    first = check(t1 , t2 , first);
    return first.getNext();
}
```

C#

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2)
{
    Node<int> first = new Node<int> (- 1);
    first = Check(t1 , t2 , first);
    return first.GetNext();
}
```

ממש את הפעולה:

ב־ Java:

```
public static Node<Integer> check( BinNode<Integer> t1,
                                BinNode<Integer> t2 , Node<Integer> list)
```

או ב־ C#:

```
public static Node<int> Check(BinNode<int> t1 , BinNode<int> t2 , Node<int> list)
```

אתה יכול להשתמש בפעולה שמימשת בסעיף א.

ג. מה היא סיבוכיות זמן הריצה של הפעולה שמימשת בסעיף ב ? נמק את תשובתך. /המשך בעמוד 9/

פרק שלישי (25 נקודות)

בפרק זה שאלות בארבעה מסלולים:
 מערכות מחשב ואסמבלי, עמודים 9-11.
 מבוא לחקר ביצועים, עמודים 12-16.
 מודלים חישוביים, עמוד 17.
 תכנות מונחה עצמים ב־ Java, עמודים 18-21; תכנות מונחה עצמים ב־ C#, עמודים 22-25.
 ענה על שאלה אחת במסלול שלמדת.

מערכות מחשב ואסמבלי

אם למדת מסלול זה, ענה על אחת מן השאלות 7-8 (25 נקודות).

7. בשאלה זו שני סעיפים, א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.
 א. לפניך קטע תכנית באסמבלי שבו ב־ AL וב־ AH יש מספרים שונים זה מזה, לא מסומנים.

```

MOV AX , 7F80H
CMP AH , AL
JL     SMALL
BIG:   MOV BL , AH
        JMP  SOF
SMALL: MOV BL , AL
SOF:
    
```

(1) מה יהיה ערכו של BL בסיום ביצוע קטע התכנית?

(2) מה מבצע קטע התכנית?

(3) בקטע התכנית, במקום ההוראה JL SMALL נכתבה ההוראה JB SMALL.

האם בסיום הביצוע של קטע התכנית לאתר החלפה יהיה ערכו של BL שונה מערכו בסיום ביצוע הקטע לפני החלפת ההוראה? הסבר את תשובתך.

ב. (אין קשר לסעיף א.)

לפניך שישה היגדים. לכל אחד מהם קבע אם הוא נכון או אינו נכון. אם ההיגד אינו נכון, הסבר מדוע.

$$10011110101110_2 > 27AE_{16} \quad (1)$$

$$27AE_{16} > 10159_{10} \quad (2)$$

(3) התפקידים של אוגרים BP ו־ SP הם:

BP קובע באיזו כתובת תִּירשם מילה במחסנית, ומאיזו כתובת במחסנית תישלף מילה.

SP מאפשר לשגרה (פרוצדורה) לשלוף פרמטרים מן המחסנית, בלי לערב את BP.

(4) בסיום הקריאה והביצוע של כל הוראה, ערכו של IP גָדל תמיד ב־ 1.

(5) לפניך שני קטעי תכנית באסמבלי.

קטע 2	קטע 1
MOV DX , 0000H	MOV DX , 0000H
MOV AX , 0064H	MOV AX , 0064H
DIV AX	DIV AL

בסיום ביצוע שני הקטעים התוכן של AX יהיה 0001H.

(6) לפניך שני קטעי תכנית באסמבלי.

קטע 2	קטע 1
MOV AL , 5BH	MOV AL , 5BH
MOV CL , 9	MOV CL , 8
ROL AL , CL	ROL AL , CL

בסיום ביצוע קטע 1 ערכו של AL יהיה שווה לערך שהוצב בו בהוראה הראשונה,

וגם בסיום ביצוע קטע 2 ערכו של AL יהיה שווה לערך שהוצב בו בהוראה הראשונה.

.8

נתונים שני מערכים ARR1 ו-ARR2. שני המערכים הם בגודל שווה וכל אחד מן האיברים שלהם הוא בגודל מילה. בכל איבר, בכל אחד מן המערכים, מאוחסן מספר הקסדיצמלי שלם בן ארבע ספרות, בתחום 1000H - FFFFH.

כמו כן נתון משתנה K בגודל בית שבו מאוחסן גודל המערכים ARR1 ו-ARR2.

כתוב באסמבלי קטע תכנית שיאחסן באוגר DX את מספר זוגות המספרים (האחד ממערך ARR1 והאחר ממערך ARR2) שמקיימים את התנאים האלה:

(i) לשני המספרים בזוג יש אותו מציין (אינדקס) במערך.

(ii) שני המספרים בזוג מורכבים מאותן ספרות אך בסדר הפוך.

דוגמה: בעבור המערכים ARR1 ו-ARR2 בגודל 5 שלפניך, יאוחסן באוגר DX המספר 2.

ARR1	2025H	1061H	1492H	5777H	1948H
ARR2	1984H	1601H	2914H	9999H	8491H

מבוא לחקר ביצועים

אם למדת מסלול זה, ענה על אחת מן השאלות 9-10 (25 נקודות).

9. נתונה בעיית תכנון לינארי:

$$\max \{z = 5x_1 - x_2\}$$

בכפוף לאילוצים האלה:

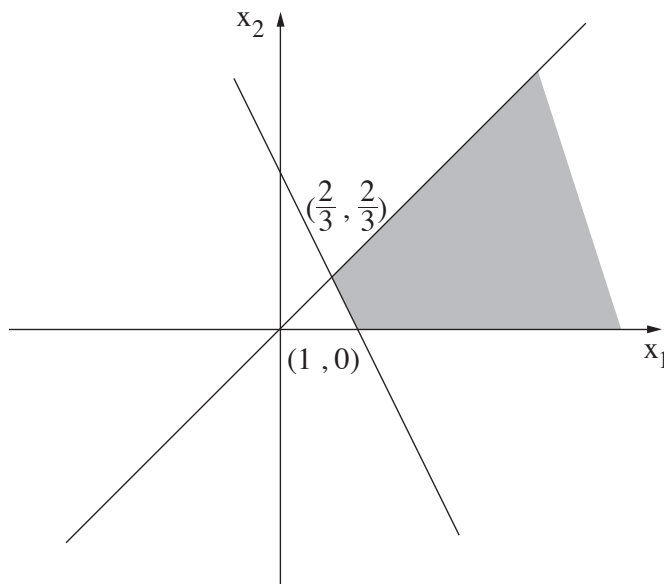
$$x_2 \leq x_1$$

$$2x_1 + x_2 \geq 2$$

$$x_1 \geq 0$$

$$x_2 \geq 0$$

לפניך סרטוט של תחום הפתרונות האפשריים של הבעיה הנתונה.



כל אחד מן הסעיפים א-ה שבעמוד הבא מתייחס לבעיית התכנון הלינארי הנתונה.

הסעיפים א-ה אינם קשורים זה לזה. ענה על כל הסעיפים.

בעבור הסעיפים א-ד נתונים ארבעה היגדים i-iv, ולכל סעיף רק היגד אחד נכון.

i הפתרון האופטימלי הוא יחיד.

ii יש אין-סוף פתרונות אופטימליים.

iii הפתרון האופטימלי לא חסום.

iv אין פתרון אופטימלי.

בכל אחד מן הסעיפים א-ד קבע איזה היגד הוא הנכון, העתק אותו למחברתך, ונמק את קביעתך.

- אם בחרת בהיגד i בסעיף כלשהו, עליך למצוא את הפתרון האופטימלי היחיד ואת הערך של פונקציית המטרה בפתרון זה.
 - אם בחרת בהיגד ii בסעיף כלשהו, עליך לרשום את הפתרון האופטימלי הכללי לבעיה ואת הערך של פונקציית המטרה בתחום הפתרונות האופטימליים.
- א. איזה היגד הוא הנכון בעבור בעיית תכנון לינארי הנתונה בתחילת השאלה? נמק את תשובתך.
- ב. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\max \{z = -5x_1 + x_2\}$. איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.
- ג. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\min \{z = -5x_1 + x_2\}$. איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.
- ד. משנים רק את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\max \{z = -2x_1 + 2x_2\}$. איזה היגד הוא הנכון לאחר שינוי פונקציית המטרה? נמק את תשובתך.
- ה. מוסיפים אילוץ נוסף לבעיה הנתונה בתחילת השאלה והוא: $2x_1 - x_2 \leq 2$ ומשנים את פונקציית המטרה של הבעיה הנתונה בתחילת השאלה ל- $\max \{z = ax_1 + x_2\}$ כאשר $-1 < a < 2$, והערך המקסימלי שמקבלת פונקציית המטרה הוא 4 בתחום הפתרונות האפשריים. חשב את הערך של a.

10. בשאלה זו שני סעיפים א-ב. אין קשר בין הסעיפים. עליך לענות על שניהם.
 א. $G = (V, E)$ הוא גרף לא מכוון המיוצג על ידי רשימת הסמיכויות שלפניך:

a	→	b	→	c	→	d	→	
b	→	a	→	c	→			
c	→	a	→	b	→			
d	→	a	→	f	→	e	→	
e	→	d	→	f	→			
f	→	e	→	d	→			

- (1) סרטט את הגרף G המיוצג על ידי רשימת הסמיכויות.
 (2) האם הגרף הנתון G הוא גרף קשיר? נמק.
 (3) הפעל אלגוריתם סריקה לעומק (DFS) על הגרף הנתון החל בקדקוד a.
 סרטט במחברתך רק את העץ הפורש (DFS) שמתקבל.
 התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.
 (4) הפעל אלגוריתם סריקה לרוחב (BFS) על הגרף הנתון החל בקדקוד a.
 סרטט במחברתך רק את העץ הפורש (BFS) שמתקבל.
 התבסס על הייצוג הנתון על ידי רשימת הסמיכויות.

ב. (אין קשר לסעיף א.)

(1) בטבלה שלפניך נתון חלק מפתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכיהם של

$$u_1, u_2, u_3, v_1, v_2, v_3.$$

מקורות	יעדים			היצע	u_i
	1	2	3		
1	2 20	5	7	20	2
2	1	1 10	4	10	0
3	0	1	8 10	15	0
ביקוש	20	15	10		
v_j	0	1	8		

העתק את הטבלה למחברתך, והשלם אותה תוך התחשבות בערכיהם של u_i -ים

ו- v_j -ים, כדי שיתקבל פתרון בסיסי אפשרי.

(שים לב: המשך הסעיף בעמוד הבא.)

(2) בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתונים ערכיהם של

$$u_1, u_2, u_3, v_1, v_2, v_3$$

מקורות	יעדים			היצע	u_i
	1	2	3		
1	14	15	17	180	0
		130	50		
2	10	8	14	100	-7
	80	20			
3	15	20	18	80	1
			80		
ביקוש	80	150	130		
v_j	17	15	17		

האם הפתרון הוא אופטימלי? נמק את תשובתך.

(3) בטבלה שלפניך נתון פתרון בסיסי אפשרי לבעיית תובלה, ונתון $v_1 = 0$.

מקורות	יעדים			היצע	u_i
	1	2	3		
1	10	25	30	20	
	20				
2	10	22	14	50	
	30		20		
3	18	20	20	60	
		40	20		
ביקוש	50	40	40		
v_j	0				

העתק את הטבלה למחברתך והשלם בה את הערכים

$$u_1, u_2, u_3, v_2, v_3$$

מודלים הישוביים

אם למדת מסלול זה, ענה על אחת מן השאלות 11-12 (25 נקודות).

11. בשאלה זו שני סעיפים א-ב. אין קשר בין הסעיפים. ענה על שניהם.

א. נתונה פעולה הכתובה ב-Java וב-C#.

Java

```
boolean foo(String str) {
    int cntA = 0;
    int cntC = 0;
    for (int i = 0; i < str.length; i++) {
        if (str[i] == 'a') cntA++;
        if (str[i] == 'c') cntC++;
    }
    if ((cntA % 2 == 0) &&
        (cntC % 3 == 0))
        return true;
    return false;
}
```

C#

```
bool Foo(string str) {
    int cntA = 0;
    int cntC = 0;
    for (int i = 0; i < str.Length; i++) {
        if (str[i] == 'a') cntA++;
        if (str[i] == 'c') cntC++;
    }
    if ((cntA % 2 == 0) &&
        (cntC % 3 == 0))
        return true;
    return false;
}
```

(1) כתוב את השפה L מעל הא"ב $\{a, c\}$ שהיא אוסף כל המילים שבעבורן

הפעולה הנתונה מחזירה true.

(2) בנה אוטומט סופי דטרמיניסטי שיקבל את השפה L .

ב. (אין קשר לסעיף א.)

בנה אוטומט סופי לא דטרמיניסטי מעל הא"ב $\{a, b\}$ שיקבל את כל המילים המכילות

לפחות מופע אחד של אחד מן הרצפים: $ababa, aaba, bbb$.

12. לפניך השפה L מעל הא"ב $\{a, b, c\}$.

$$L = \{a^n b^m c^{n+m} \mid n, m > 0\}$$

א. בנה אוטומט מחסנית שיקבל את השפה L .

ב. בנה מכונת טיורינג שתקבל את השפה L .

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב־ Java, ענה על אחת מן השאלות 13-14. (25 נקודות)
 13. לפניך הממשקים IOne, ITwo, IThree, IFour והמחלקות Alpha, Beta, Gamma, IOne ו־ Program.

```
public interface IOne {
    public boolean firstA(Object x);
    public void firstB(int num);
}
```

```
public interface ITwo {
    public int second();
}
```

```
public interface IThree {
    public int third();
}
```

```
public interface IFour {
    public int fourth();
}
```

```
public class Alpha implements IOne {}
public class Beta implements ITwo, IFour {}
public class Gamma implements ITwo, IThree {}
public class Program {
    public static void main(String[] args) {}
}
```

א. כתוב את שמות הפעולות שיש לממש בכל אחת מן המחלקות Alpha, Beta, Gamma, והסבר מדוע.

הנח שבכל אחת מן המחלקות מומשו הפעולות שכתבת בסעיף א.

ב. האם ההגדרה `public class Omega implements IFour extends Beta { }` תקינה?
אם היא אינה תקינה – הסבר מדוע.

ג. לכל אחד מן הקטעים i-iv שלפניך, אם נכתוב אותו בפעולה main – קבע אם הפעולה main שתתקבל היא תקינה או אינה תקינה. נמק את קביעותיך.

- i `ITwo x1 = new ITwo();`
- ii `Beta b = new Beta();`
- iii `Alpha a = new Alpha();`
`IOne x2 = a;`
- iv `Gamma c = new Gamma ();`
`IOne x3 = c ;`

ד. לכל אחת מן הדרישות ii-i שלפניך, קבע אם אפשר לבצע אותה באמצעות כתיבת הוראה או הוראות בפעולה main. אם אפשר – כתוב את ההוראה המתאימה או את ההוראות המתאימות. אם אי-אפשר – הסבר מדוע.

- i הפעלת הפעולה second על עצם מטיפוס Beta.
- ii המרת עצם מטיפוס Gamma שיהיה עצם מטיפוס Alpha.

14. בשאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עליך לענות על כל הסעיפים.

א. נתונות המחלקה A והמחלקה B היורשת מ־ A .

ההוראה $A\ a1 = \text{new } B()$; עוברת הידור (קומפילציה) ורצה בצורה תקינה.

בעבור כל אחד מן המשפטים iii-i שלפניך, קבע אם המשפט נכון או אינו נכון. הסבר את קביעותיך.

i אי־אפשר לכתוב את ההוראה $\text{Object } obj = a1$, מכיוון שדרושה המרה מפורשת (casting) .

ii אי־אפשר לכתוב את ההוראה $A\ a2 = a1$, מכיוון שדרושה המרה מפורשת (casting) .

iii אי־אפשר לכתוב את ההוראה $B\ b1 = a1$, מכיוון שדרושה המרה מפורשת (casting) .

ב. נתונה הפעולה main במחלקה Program :

```
public static void main(String[] args) {
    C c = new A();
    B b1 = (B) (new A());
    B b2 = new D();
    A a = new D();
}
```

קבע איזו מן האפשרויות v-i שלפניך מתארת קשר בין המחלקות, לפי הפעולה main הנתונה.

i C יורשת מ־ A , B יורשת מ־ A , D יורשת מ־ A .

ii A יורשת מ־ C , B יורשת מ־ A , D יורשת מ־ A .

iii C יורשת מ־ A , B יורשת מ־ A , D יורשת מ־ B .

iv A יורשת מ־ C , C יורשת מ־ B , D יורשת מ־ A .

v A יורשת מ־ C , B יורשת מ־ A , D יורשת מ־ B , C יורשת מ־ B .

ג. נתונות הגדרות של המחלקות School ו־ City :

```
public class City{____ (1) ____ int students; }
public class School extends City {
    void poll() { System.out.println("There are "+ students + " in the school ") ; }
}
```

כדי שלא תהיה שגיאת הידור (קומפילציה) אפשר להשלים את חלק השורה המסומן (1) באמצעות אחת מן האפשרויות שלפניך. בחר באפשרות המתאימה והעתק אותה למחברתך.

- לכתוב public , private , או protected .
- לכתוב רק public או private .
- לכתוב רק public או protected .
- לכתוב רק protected .
- לכתוב רק public .
- לכתוב רק private .

ד. לפניך הגדרה של המחלקה Quest :

```
public class Quest {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }

    public void printNow()
    {
        System.out.println (num + "" + count);
    }
}
```

קבע כמה עצמים (אובייקטים) יש לייצר מן המחלקה Quest, ועל איזה מהם יש להפעיל את הפעולה printNow, כדי שהפלט יהיה 35.

תכנות מונחה עצמים

אם למדת מסלול זה ואתה כותב ב- C#, ענה על אחת מן השאלות 15-16. (25 נקודות)

15. לפניך הממשקים IOne, ITwo, IThree, ו-IFour והמחלקות Alpha, Beta, Gamma, ו-Program.

```
interface IOne {
    bool FirstA(Object x);
    void FirstB(int num);
}

interface ITwo {
    int Second();
}

interface IThree {
    int Third();
}

interface IFour {
    int Fourth();
}

public class Alpha : IOne {}
public class Beta : ITwo, IFour {}
public class Gamma : ITwo, IThree {}
public class Program {
    public static void Main() {}
}
```

א. כתוב את שמות הפעולות שיש לממש בכל אחת מן המחלקות Alpha, Beta, ו- Gamma, והסבר מדוע.

הנח שבכל אחת מן המחלקות מומשו הפעולות שכתבת בסעיף א.

ב. האם ההגדרה `public class Omega : IFour, Beta { }` תקינה?

אם היא אינה תקינה – הסבר מדוע.

ג. לכל אחד מן הקטעים i-iv שלפניך, אם נכתוב אותו בפעולה Main – קבע אם

הפעולה Main שתקבל היא תקינה או אינו תקינה. נמק את קביעותיך.

i `ITwo x1 = new ITwo();`

ii `Beta b = new Beta();`

iii `Alpha a = new Alpha();`

`IOne x2 = a;`

iv `Gamma c = new Gamma ();`

`IOne x3 = c ;`

ד. לכל אחת מן הדרישות ii-i שלפניך קבע אם אפשר לבצע אותה באמצעות כתיבת

הוראה או הוראות בפעולה Main. אם אפשר – כתוב את ההוראה המתאימה או את

ההוראות המתאימות. אם אי-אפשר – הסבר מדוע.

i הפעלת הפעולה Second על עצם מטיפוס Beta.

ii המרת עצם מטיפוס Gamma שיהיה עצם מטיפוס Alpha.

16. בשאלה זו ארבעה סעיפים, א-ד. אין קשר בין הסעיפים. עליך לענות על כל הסעיפים.

א. נתונות המחלקה A והמחלקה B היורשת מ-A.

ההוראה: `A a1 = new B();` עוברת ההידור (קומפילציה) ורצה בצורה תקינה.

בעבור כל אחד מן המשפטים iii-i שלפניך, קבע אם המשפט נכון או אינו נכון. הסבר את קביעותיך.

i אי-אפשר לכתוב את ההוראה: `Object obj = a1;`, מכיוון שדרושה המרה מפורשת (casting).

ii אי-אפשר לכתוב את ההוראה: `A a2 = a1;`, מכיוון שדרושה המרה מפורשת (casting).

iii אי-אפשר לכתוב את ההוראה: `B b1 = a1;`, מכיוון שדרושה המרה מפורשת (casting).

ב. נתונה הפעולה Main במחלקה Program:

```
public static void Main(string[] args) {
    C c = new A();
    B b1 = (B) (new A());
    B b2 = new D();
    A a = new D();
}
```

קבע איזו מן האפשרויות v-i שלפניך מתארת קשר בין המחלקות, לפי הפעולה Main הנתונה.

i C יורשת מ-A, B יורשת מ-A, D יורשת מ-A.

ii A יורשת מ-C, B יורשת מ-A, D יורשת מ-A.

iii C יורשת מ-A, B יורשת מ-A, D יורשת מ-B.

iv A יורשת מ-C, C יורשת מ-B, D יורשת מ-A.

v A יורשת מ-C, B יורשת מ-A, D יורשת מ-B, C יורשת מ-B.

ג. נתונות הגדרות המחלקות City ו- School :

```
public class City { ____ (1) ____ int students; }
public class School : City {
    void Poll() { Console.WriteLine("There are "+ students + " in the school " ); }
}
```

כדי שלא תהיה שגיאת הידור (קומפילציה) אפשר להשלים את חלק השורה המסומן (1) באמצעות אחת מן האפשרויות שלפניך. בחר באפשרות המתאימה והעתק אותה למחברתך.

- לכתוב public , private , או protected .
- לכתוב רק public או private .
- לכתוב רק public או protected .
- לכתוב רק protected .
- לכתוב רק public .
- לכתוב רק private .

ד. לפניך הגדרה של המחלקה Quest :

```
public class Quest {
    private int num;
    private static int count = 0;

    public Quest()
    {
        count++;
        num = count;
    }

    public void PrintNow()
    {
        Console.WriteLine(num+" "+count);
    }
}
```

קבע כמה עצמים (אובייקטים) יש לייצר מן המחלקה Quest , ועל איזה מהם יש להפעיל את הפעולה PrintNow , כדי שהפלט יהיה 35 .

בהצלחה!